Pattern Recognition in Multivariate Time Series

[Thesis Proposal]

Stephan Spiegel *, Brijnesh-J. Jain †, Ernesto W. De Luca ‡, Sahin Albayrak §
DAI-Labor
Technische Universitaet Berlin
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
{stephan.spiegel,brijnesh-johannes.jain,ernesto.deluca,sahin.albayrak}@dai-labor.de

ABSTRACT

Nowadays computer scientists are faced with fast growing and permanently evolving data, which are represented as observations made sequentially in time. A common problem in the data mining community is the recognition of recurring patterns within temporal databases or streaming data. This dissertation proposal aims at developing and investigating efficient methods for the recognition of contextual patterns in multivariate time series in different application domains based on machine learning techniques. To this end, we propose a generic three-step approach that involves (1) feature extraction to build robust learning models based on significant time series characteristics, (2) segmentation to identify internally homogeneous time intervals and change points, as well as (3) clustering and/or classification to group the time series (segments) into the sub-population to which they belong to. To support our proposed approach, we present and discuss first experiments on real-life vehicular data. Furthermore we describe a number of applications, where pattern recognition in multivariate time series is practical or rather necessary.

Categories and Subject Descriptors


General Terms

Algorithm, Design, Experimentation

*Stephan Spiegel is the author of this Ph.D. proposal.
†Brijnesh-Johannes Jain is the second research advisor.
‡Ernesto William De Luca is the main research advisor.
§Sahin Albayrak is the first supervisor the proposed thesis.

Keywords

Feature Extraction, Time Series Segmentation, Clustering and Classification, Machine Learning Algorithms

1. INTRODUCTION

Since most electronic components have gotten substantially smaller and cheaper to produce, an increasing number of everyday electronic devices are equipped with smart sensors. Recent mobile phones have integrated location and acceleration sensors [7], modern vehicles are able to measure ambient temperature and average fuel consumption [36], and twenty-first-century smart homes monitor user activities as well as consumption of resources [32]. There is a significant trend in using these integrated sensors to simplify human-computer-interaction by means of automatic adaptation of user interfaces or even machine behavior according to the recognized context.

According to the Oxford Dictionary the term "context" is defined as "the circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood and assessed". Although the term has been used in many ways in different areas of computer science [7, 10], we only focus on the context that can be discovered from sensor data, which are basically patterns that represent rare events (anomalies) or recurring situations. Recently, there has been done a lot of research in the field of context-aware systems, ranging from adaptive applications for (mobile) electronic devices and ubiquitous applications in smart home environments to advanced driver assistance systems (ADAS) and event detection in video sequences. Context-aware applications take advantage of environmental characteristics to adapt to user needs without consuming user attention [7, 10, 26]. Smart environments make it possible to build ubiquitous applications that assist users during their everyday life, at any time, in any context [23, 32, 34]. Leading car manufacturers like the Volkswagen AG aim to identify characteristic drive(r) profiles from abnormal and recurring context (i.e. motor behavior and drive maneuvers) by analyzing vehicular sensor data recorded during car drives [36]. Media and entertainment companies like Technicolor push violent scene detection in videos [4, 11, 12] to help users to choose movies that are suitable for their children of different age: where corresponding context or rather events are discovered from audio and video features represented as changing signals equal to sensor data. All these application examples involve large amounts of high dimensional data with significant between-sensor correla-
tions that are corrupted by non-uniform noise. To deal with these problems, we adopt a machine learning approach for detecting contextual patterns in multivariate time series [36]. With regard to sensor data, the term multivariate implies the exploration of multiple signals in parallel. We especially focus on multivariate time series analysis, because almost all real-life applications employ multiple sensors to retrieve contextual patterns, which describe environmental situations or system states.

Our research goal follows a dual agenda: First, we develop efficient methods for detecting contextual patterns in multivariate time series based on state-of-the-art and recent machine learning techniques. Second, we focus on developing methods that are not tailored to solutions of specific application areas, but that will be applicable to other domains as well. To achieve this, we suggest a generic approach for contextual pattern detection in sensor data consisting of the following three steps: feature extraction, segmentation and clustering/classification.

Since multi-sensor data is often too complex and highly redundant [19], we transform the raw input data into a reduced representation in the feature extraction step. The challenge consists in extracting features in such a way that the reduced data contains all or most of the relevant information [14] for the underlying task of detecting contextual patterns. Based on the selected signal features we can partition a time series into segments or situations [1, 2], which might be classified or clustered to high-level context in a second step. This paper introduces straight-forward signal processing features as well as more sophisticated feature extraction algorithms to measure similarity between two time series (segments). Given a feature representation of the input data, segmentation aims at partitioning the time series into internally homogeneous intervals of constant correlation. Our underlying assumptions are that contextual patterns arise from correlations of the given sensor data [36]. With regard to sensor signals, context or events are usually described as segments of multivariate time series, which are internally homogeneous. In the domain of time series analysis the term homogeneous refers to an unchanging correlation between the observed variables over a specified period of time. Due to the fact that most situations have different length and are often overlapping, it is a non-trivial task to determine clear borders between individual time series segments [1, 2]. Depending on the application and the examined dataset, we either want to group the retrieved time series segments into clusters of similar objects or we aim to classify the identified sub-populations to which they belong. A preliminary evaluation on contextual pattern recognition in vehicular sensor data is presented in Section 6. Our research ideas that can mature into a dissertation are discussed in Section 7.

2. PROBLEM STATEMENT

"Intelligence is the ability to adapt to change" - Stephen Hawking. With regard to machine intelligence, one could argue that a machine needs to be aware of its operating context to adapt to change and to be considered as intelligent. Most intelligent systems employ smart sensors to infer context, which can be used to adapt to the (environmental or system) changes at hand. Our goal is to recognize contextual patterns in multivariate time series, which can be used as an information source for intelligent systems to adapt to their ever-changing environment.

For further problem discussion we represent sensor data as a multivariate time series \( T = (v_{ij}) \in \mathbb{R}^{m \times N} \) where \( m \) is the number of measurements and \( N \) is the number of discrete time points \( t_1, \ldots, t_N \) at which the measurements are taken. A contextual pattern (i.e. event or situation) of \( T \) is a pair \( p = (s, c) \) consisting of a segment \( s = (x_i, x_{i+1}, \ldots, x_{i+j}) \) of consecutive columns of \( T \) together with a class label \( c \) from some set \( C = \{c_1, \ldots, c_n\} \). A segment \( s \) describes the characteristic features of a contextual pattern and determines its period. The class label \( c \) describes the nature of the

Figure 1: Segmentation of speed and pedal signal
This plot illustrates a straight-forward time series segmentation according to the critical points (i.e. extrema) of the filtered and smoothed (Savitzky-Golay Filter) speed signal.
contextual pattern. Detecting contextual patterns amounts in finding a column-wise segmentation \( \mathcal{S} = \{ s_1, \ldots, s_q \} \) of the matrix \( T \) together with a labeling function \( \ell : \mathcal{S} \rightarrow \mathcal{C} \), \( s_i \mapsto \ell(s_i) \) that assigns a class label to each segment. Here, we demand that a segmentation \( \mathcal{S} \) of \( T \) consists of nonempty sets \( s_i \) of consecutive columns that form a partition of the column set of \( T \).

It is important to note that \( \mathcal{C} \) can be unknown and therefore constitutes a part of the learning problem. Thus, detecting contextual patterns in multivariate time series amounts in learning both, the data (segments) we want to classify as well as the class labels (contexts) we want to assign to the data. This is in contrast to traditional machine learning, where the class labels as well as the data we want to learn on are usually given.

3. THREE-STEP APPROACH

The proposed three-step approach for contextual pattern recognition in multivariate time series involves feature extraction, segmentation and clustering/classification. Each of the three subtasks can be fulfilled by different techniques, some of them are discussed in the following sections. This treatment also includes own contributions to underpin the thesis proposal.

3.1 Feature Extraction

A time series is a collection of observations made sequentially in time. People measure things, and things (with rare exception) change. In most cases, patterns, and not individual points, are of interest [19]. Time series collections are often huge and mining them requires a high-level representation of the data. Due to the fact that time series analysis aims to be time and space efficient, data mining researchers usually extract significant features or characteristics which given an approximation of the original data.

Feature extraction involves simplifying the amount of resources required to describe a large set of data accurately. When performing analysis of complex data one of the major problems stems from the number of variables involved. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy [16].

In machine learning and statistics, feature selection is the technique of selecting a subset of relevant characteristics for building robust learning models. From a theoretical perspective, it can be shown that optimal feature selection for (un-)supervised learning problems requires an exhaustive search of all possible subsets of features [14]. For practical learning algorithms, the search is for a satisfactory set of features instead of an optimal set.

In the following we discuss several well-known signal processing features that are employed to describe time series. Furthermore we go into different similarity measures that are used to discriminate time series, which is necessary for clustering and classification.

Fourier Transform (FT) is one of the most popular signal processing features and a measure of how much of each frequency is existing in the signal [22]. If the time domain of a signal is given and the frequency amplitude is searched then the Fourier Transform is used. In general, FT is only valid for periodical signals. A special case of the continuous FT is the Discrete Fourier Transform (DFT), which requires a discrete periodical spectrum defined on a certain area. Therefore a sample rate needs to be defined.

Sometimes it is necessary to have both time and frequency information simultaneously, so a time localization of the spectral components is needed. Discrete Wavelet Transform (DWT) is a technique that decomposes a signal into several groups (vectors) of coefficients. Different coefficient vectors contain information about characteristics of the sequence at different scales [13].

Mel Frequency Cepstral Coefficients (MFCCs) are the dominant feature in speech recognition [25]. MFCCs are short-term spectral features that are calculated in several steps. First the signal is divided into frames. For each frame, the algorithm obtains the amplitude spectrum, which in turn is converted to Mel (a perceptually-based) spectrum, so called Mel scale. This transformation emphasizes lower frequencies which are perceptually more meaningful for speech. It is possible however that the Mel scale may not be optimal for music as there may be more information in say higher frequencies. Finally the algorithm takes the discrete cosine transform (DCT) to decor-relate the components of the feature vectors. MFCC’s might be relevant for event detection in video sequences as well.

Principal Component Analysis (PCA) is a matrix factorization technique which is used to reduce the dimensionality of the input space and/or to retrieve latent relations between variables of the observed dataset. PCA can be found in many applications, such as recommender systems [20, 33, 38], semantic analysis of textual information [9] and link prediction in complex networks [3, 24, 37]. Furthermore, PCA is a common feature extraction method in signal processing, which is employed for time series segmentation and clustering [1, 2, 36]. The sensor fusion algorithm developed by Abonyi et al. [1, 2] is able to segment a multivariate time series in a bottom-up manner. Adjacent segments are merged iteratively if their combined model does not exceed a predefined reconstruction error, which is computed by means of principal component analysis. One major advantage of the proposed segmentation algorithm is that it allows one to reuse the PCA model to define a distance measure to compare multivariate time series segments for clustering or classification [2, 35]. Note that PCA-based algorithms are able to detect changes in the mean, variance and correlation structure among multiple variables or signals [36].

In the context of pattern recognition in multivariate time series, feature extraction can be considered as a preprocessing step for further data mining and machine learning algorithms. However, for the clustering and classification task it is necessary to define a distance measure to compare time series (segments). Usually the established feature vectors are compared by means of cosine similarity or other metrics that conform to the euclidean space. Apart from that, there exist distance measures that work on the raw sensor data or time series measurements.

The Dynamic Time Warping (DTW) distance measure is a technique that has long been known in speech recognition community [30]. It allows a non-linear mapping of one signal to another by minimizing the distance between the two. A decade ago, DTW was introduced to the data mining community as a utility for various tasks for time series problems including clustering, classification, and anomaly detection [6]. Although, DTW is superior to Euclidean Distance (ED) for clustering and classification of time series,
most research has utilized ED because it is more efficiently calculated [31]. Lower bounding (LB) that greatly mitigates DTWs demanding CPU time has sparked a flurry of research activity. However, LB and its applications still only allow DTW to be applied to moderate large datasets. In addition, almost all research on DTW has focused on speeding up calculations; there has been little work done on improving its accuracy.

### 3.2 Time Series Segmentation

Segmentation is the most frequently used subroutine in both clustering and classification of time series. It is used to locate stable periods of time or alternatively to identify change points [2]. There already exist different heuristic approaches to extract internally homogeneous segments. Some primitive algorithms search for inflection points to locate episodes [39]. Other algorithms determine segment borders by the Sliding Window technique, where a segment is grown until it exceeds some error bound [19].

In our previous work [36] we proposed a bottom-up segmentation approach, which begins creating a fine approximation of the time series, and iteratively merges the lowest cost pair of segments until some stopping criteria is met. The cost of merging two adjacent segments is evaluated by the Singular Value Decomposition (SVD) model of the new segment. SVD-based algorithms are able to detect changes in the mean, variance and correlation structure among several variables [37, 38, 36]. The proposed approach can be considered as an extension of the sensor fusion algorithm developed by Abonyi [1, 2].

For the purpose of finding internally homogeneous segments from a given time series, we need to formalize a cost function for the individual time intervals. In most cases, the cost function is based on the distance between the actual values of the time series and a simple function (linear function, or polynomial of higher degree) fitted to the data of each segment [2]. Since our approach aims at detecting changes in the correlation structure among several variables, the cost function of the segmentation is based on the Singular Value Decomposition of the segment matrices, where each row is an observation, and each column is a variable. Before applying SVD, the observed variables need to be centered and scaled, in such a way as to make them comparable. Hence, we compute the z-scores using mean and standard deviation along each individual variable of the time series.

The SVD model projects the correlated high-dimensional data onto a lower-dimensional hyperplane which is useful for the analysis of multivariate data. The distance of the original data from the hyperplane is a significant indicator for anomalies or unsteadiness in the progression of the observed variables. Hence, it is useful to analyze the reconstruction error of the SVD model to get information about the homogeneity of the factorized time series segments. In the proposed approach the reconstruction error is determined by the Q-measure [2, 36], which is commonly used for the monitoring of multivariate systems and for the exploration of the errors. The crux of the proposed time series segmentation is to use the Q-measure as an indicator for the homogeneity of individual or rather merged segments.

The sensor fusion algorithm developed by Abonyi [2] merges segments bottom-up, whereas the initial fine-grain segmentation of the time series is determined manually. In case of a periodic signal and an odd initial segmentation it is highly probable that the bottom-up algorithm will not find recurring segments, because the time series was partitioned into internal inhomogeneous time intervals. Therefore, we propose to perform an initial fine-grain segmentation of the time series according to the critical points of the individual signals. Critical points of great interest are extrema as well as inflection and saddle points [39]. In order to determine the critical points of a curve or signal we need to calculate the first, second or third derivative respectively.

Due to the fact that sensors often produce extremely noisy and fluctuate signals, it is very likely that the examined time series exhibit dozens of critical points. Hence, we need to smooth the signal with a low-pass or base-band-pass filter that reduces the number of critical points, but does not substantially change the coordinates of the segmentation borders. We propose to employ the Savitzky-Golay filter [27], which is typically used to smooth out a noisy signal whose frequency span is large. Savitzky-Golay smoothing filters perform much better than standard averaging FIR (Finite Impulse Respond) filters, which tend to filter out a significant portion of the signal's high frequency content along with the noise. Although Savitzky-Golay filters are more effective at preserving the pertinent high frequency components of the signal, they are less successful than standard averaging FIR filters at rejecting noise. Savitzky-Golay filters are optimal in the sense that they minimize the least-squares error in fitting a higher-order polynomial to frames of noisy data [27]. The introduced critical point approach can be employed as a preprocessing step for the proposed bottom-up segmentation, but can also be considered as a segmentation technique by its own.

### 3.3 Clustering and Classification

Depending on the way of looking at the stated pattern recognition problem we either want to classify or cluster the time series (segments). Classification is a supervised learning technique, which employs labeled training data to gradually adjust model parameters until the classifier fits the presented data. The trained model can then be employed to predict labels for previously unseen input or time series (segments) respectively. In contrast, clustering is a method of unsupervised learning, which is used to group similar objects without training the mathematical model. Cluster analysis gives information about the underlying structure of the examined data and can be employed to identify recurring time series segments that fall into the same group and unique time intervals that are represented as outliers.

Hierarchical clustering methods find successive groups using previously established ones. Agglomerative hierarchical clustering algorithms begin with each element as a separate cluster and merge them into successively larger clusters. Merges and splits are determined in a greedy manner. The results of hierarchical clustering are usually presented as a dendrogram. Hierarchical clustering has the distinct advantage that any valid measure of distance can be used. In fact, the observations themselves are not required; all that is used is a matrix of distances.

The choice of an appropriate metric does influence the shape of the clusters, as some elements might be close to each other according to one distance and farther away according to another. Some commonly used metrics for hierarchical clustering are Euclidean distance, cosine similarity, Mahalanobis distance (covariance matrix) and Pearson’s correlation. For
text or other non-numeric data, metrics such as the Hamming distance or Levenshtein distance are often used. Given an appropriate metric, an agglomerative hierarchical cluster tree can be created by several different methods, which differ from one another in how they measure the distance between the clusters. Available methods are single and complete linkage, also known as shortest and furthest distance, as well as average, centroid and inner-squared distance.

In the following we want to give a brief introduction to several popular classification techniques, which can be used to categorize time series (segments). The k-nearest neighbor (k-NN) algorithm [8] is amongst the simplest of all machine learning approaches. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k-nearest neighbors. If \( k = 1 \), then the object is simply assigned to the class of its nearest neighbor (also known as 1-NN classifier).

Another popular classification technique in the machine learning community are decision trees, which are tree-like graphs of decisions and their possible consequences [29]. In data mining, trees can be described as the combination of mathematical and computational techniques to aid the description, categorization and generalization of a given set of data. They are used as a predictive model which maps observations about an item to conclusions about the item’s target value. A tree can be learned by splitting the source set into subsets based on an attribute value test.

In addition, there also exist probabilistic models like Bayesian networks, which represent a set of random variables and their conditional dependencies via a directed acyclic graph [15]. For example, a Bayesian network could represent the probabilistic relationships between complex situations and corresponding sensor data. One advantage of Bayesian networks is that it is intuitively easier for a human to understand direct dependencies and local distributions than complete joint distribution. Furthermore, Bayesian Networks can save considerable amount of memory compared to naive matrix based storing of conditional dependencies, if the dependencies in the joint distribution are sparse.

Recently, there has been done a lot of research in the field of kernel method, especially Support Vector Machines (SVM) [17, 28]. Given a set of training samples, each labeled as belonging to one of two categories, a SVM training algorithm builds a model that predicts whether a new sample falls into one category or the other. Intuitively, a SVM model is a representation of the samples as points in space, mapped so that the samples of the separate categories are divided by a clear margin that is as wide as possible. New samples are then mapped into that same space and predicted to belong to a category based on which side of the margin they fall. More formally, a SVM constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. In general, the effectiveness of a SVM lies in the selection of the soft margin parameters and the kernel, which is a weighted function used in non-parametric estimation techniques.

4. EVALUATION ON VEHICULAR DATA

We applied the proposed three-step approach to the recognition of complex drive maneuvers in real-life vehicular sensor data recorded during car drives. Although many different environmental and motor parameters were measured, we merely focus on the progressing of speed, revolution, gear and accelerator signal, which are highly correlated and provide sufficient informative value to perform a preliminary evaluation. We refer to the technical paper [36] for a more detailed description of the experimental settings.

Our proposed pattern recognition approach [36] extends the sensor fusion algorithm developed by Abonyi [2]. We use PCA-based bottom-up segmentation to partition the time series data into homogeneous intervals that can be viewed as situations, and subsequently group the recognized segments by means of agglomerative clustering using straightforward statistical features, such as standard deviation, local extrema as well as the slope of the individual signals within a segment. For the clustering task we employ cosine similarity to calculate the distances between feature vectors and average linkage to compute average distance between all pairs of objects (i.e. segments) in any two groups.

The question is: what makes up a “good” segment and at which point of our bottom-up algorithm we should stop to merge adjacent segments? We are able to determine the merge threshold by means of the applied cost function, that is the sum of the Q-measures over all segments [2, 36]. A possible termination criteria for merging time series segments could be a steep ascent of the cost function. Beside automatically determining the number of segments (merge threshold), we furthermore compared different approaches of time series segmentation by evaluating the overall cost of the established segments. Table 1 shows a comparison of our proposed bottom-up segmentation (BU), the introduced critical point (CP) approach and a mixed BUCP segmentation for several different car drives.

<table>
<thead>
<tr>
<th>Car Drive</th>
<th>BU Q-Measure</th>
<th>CP Q-Measure</th>
<th>BUCP Q-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car Drive-1</td>
<td>1.04</td>
<td>88.84</td>
<td>88.84</td>
</tr>
<tr>
<td>Car Drive-2</td>
<td>79.45</td>
<td>375.24</td>
<td>375.13</td>
</tr>
<tr>
<td>Car Drive-3</td>
<td>61.94</td>
<td>305.08</td>
<td>305.06</td>
</tr>
<tr>
<td>Car Drive-4</td>
<td>32.57</td>
<td>181.31</td>
<td>180.97</td>
</tr>
</tbody>
</table>

Table 1: Performance of Segmentation Algorithms
Note: The lower the Q-measure the better [36].
the hierarchical cluster tree generated during segment grouping. We worked out that the slope of the distance function is an useful indicator for determining a reasonable number of clusters with sufficient differentiation [36]. However, cluster quantity always depends on the underlying structure of the time series and might vary from dataset to dataset. Although we discussed pattern recognition in terms of sensor data recorded from vehicles, the introduced three-step time series analysis is applicable to datasets from other domains as well. The following section affords a brief outlook on our future work and presents research ideas that may mature into a dissertation.

5. FUTURE WORK

In our future work, we plan further analysis of vehicular sensor data to categorize drives and drivers. Usually, drive maneuvers differ from person to person, and also depend on the motorization of the car. Therefore, it is interesting to analyze the patterns that characterize individuals or vehicles respectively. This knowledge is especially valuable for car manufacturers like the Volkswagen AG, because it allows more efficient testing of recently developed engines [36].

For the classification of contextual patterns in car driving we plan to employ the Dynamic Time Warping (DTW) distance measure [30, 31], because it is able to handle local scaling (warping) invariance [5]. Kroschel et al. [21] describe an advanced DTW segmentation and classification approach, which detects change points and categorizes time series segments simultaneously. We aim to extend this approach to cope with the task of multivariate time series analysis. Since our goal is to design solutions for diverse application scenarios, our long term perspective consists in exploring feature extraction, segmentation and clustering/classification methods for contextual pattern recognition in sensor data from other domains, such as smart home environments and mobile electronic devices. We expect to gain insight in how selected machine learning methods cooperate in multi-step approaches for solving the problem of detecting contextual patterns in multivariate time series.

6. ACKNOWLEDGMENTS

The proposed pattern recognition and classification for multivariate time series [36] was developed in cooperation with the Volkswagen AG, Wolfsburg. Thanks to Bernd Werther and Matthias Pries for providing sensor data from vehicles, and for their contribution of expert knowledge.

7. REFERENCES

USA, 2008. World Scientific and Engineering Academy and Society (WSEAS).


[38] S. Spiegel, J. Kunegis, and F. Li. Hydra: a hybrid recommender system [cross-linked rating and content information]. In CIKM-09: Proceeding of the 1st ACM international workshop on Complex networks meet information and knowledge management (CNIKM-09), pages 75–80, New York, NY, USA, 2009. ACM.


APPENDIX

A. TIME SERIES REPOSITORY

Keogh et al. have created a comprehensive time series repository 1 for the data mining and machine learning community, to encourage reproducible research for time series clustering and classification. Amongst others their current research includes similarity searching in sequence databases, local adaptive classification techniques and analysis of sensor data [5, 18, 19, 30, 31].

1http://www.cs.ucr.edu/~eamonn/time_series_data/